

APPARATUS AND METHOD FOR AUTOMATIC POLARITY SWAP IN A COMMUNICATIONS SYSTEM

Inventor: Vasudevan Parthasarathy

BACKGROUND OF THE INVENTION

Field of the Invention

- [1] The present invention generally relates to serial de-serializer integrated circuits with multiple high speed data ports, and more particularly to an automatic polarity swap to correct for any differential cross-connect at the input of one or more serial data ports.

Background Art

- [2] High speed data links transmit data from one location to another over transmission lines. These data links can include serial deserializer data links (i.e. SERDES) that receive data in a parallel format and convert the data to a serial format for high speed transmission. SERDES data links can be part of a backplane in a communications system (e.g. Tyco Backplane 30-inch trace).
- [3] In a high speed back plane configuration, it is often desirable to switch between multiple Serdes links. In other words, it is often desirable to switch between any one of multiple Serdes links to another Serdes link, and to do so in a low power configuration on a single integrated circuit.

BRIEF SUMMARY OF THE INVENTION

- [4] A multi-port Serdes transceiver includes multiple parallel ports and serial ports, and includes the flexibility to connect any one of the parallel ports to another parallel port or to a serial port, or both. Furthermore, the multiport transceiver chip can connect any one of the serial ports to another serial port or to one of the parallel ports. The multi-port Serdes transceiver is able to operate at multiple data rates.
- [5] The multi-port Serdes transceiver also includes a packet bit error rate tester (BERT). The packet BERT generates and processes packet test data that can be transmitted over any of the serial ports to perform bit error testing. The packet BERT can monitor (or "snoop") between the serial ports. In other words, if data is being transmitted from one serial port to another serial port, the packet BERT can capture and store a portion of this data for bit error testing.
- [6] The substrate layout of the multi-port Serdes transceiver chip is configured so that the parallel ports and the serial ports are on the outer perimeter of the substrate. A logic core is at the center of the substrate, where the logic core operates the serial and parallel data ports, and a bus that connects the data ports. The bus can be described as a "ring" structure (or donut "structure") around the logic core, and is configured between the logic core and the data ports. The ring structure of the bus provides efficient communication between the logic core and the various data ports.
- [7] The Serdes transceiver described herein is highly flexible and can be configured to provide multiple different transceiver products from enabling and disabling various serial and parallel data ports. This is accomplished using a configuration logic circuit that enables/disables these data ports. As a result, several different transceiver products, with different capabilities and price points, can be configured from a single semiconductor die.

[8] In one embodiment, two or more transceivers having differential inputs and outputs can be coupled together through an interface, such as a backplane to form a communications system. In such a configuration, it is possible to cross-connect the differential data lines or signals at the interface, which will cause invalid data words to be received at the second transceiver. Accordingly, the present invention includes an error check and correction module that detects invalid data words after parallel-to-serial conversion. More specifically, an error check determines if the parallel differential signal represents a valid data word. This can be done, for example, by storing and comparing valid data words in a memory such as RAM. If the received data word is valid, then no corrective action is taken. However, if the received data word is invalid, then the parallel differential signal is inverted using a logic circuit, which will correct the error if it is due to cross-connection of the differential lines.

[9] In one embodiment, the inversion of the parallel differential signal is implemented using an X-OR logic circuit. The X-OR logic circuit operates on the parallel differential signal according to a control bit. When the control bit is set to "1", the X-OR logic circuit will invert the signal components of the parallel differential signal. Other logic circuits could also be used to perform the signal inversion.

[10] Further features and advantages of the present invention, as well as the structure and operation of various embodiments of the present invention, are described in detail below with reference to the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[11] The present invention is described with reference to the accompanying drawings. In the drawings, like reference numbers indicate identical or functionally similar elements. Additionally, the left-most

digit(s) of a reference number identifies the drawing in which the reference number first appears.

[12] FIG. 1 illustrates a multi-port Serdes chip according to embodiments of the present invention.

[13] FIG. 2 further illustrates the multi-port Serdes chip including a parallel-to-serial conversion according to embodiments of the present invention.

[14] FIG. 3 further illustrates the operation of the transceiver 100 in the routing of serial data between ports according to embodiments of the present invention.

[15] FIG. 4 illustrates a multi-port Serdes transceiver 400, which is one embodiment of the transceiver 100.

[16] FIG. 5 illustrates a substrate layout of the multi-port Serdes transceiver chip according to embodiments of the present invention.

[17] FIG. 6 illustrates a section of the bus 106 according to embodiments of the present invention.

[18] FIG. 7 further illustrates the bus 106 having equal length transmission lines according to embodiments of the present invention.

[19] FIG. 8 illustrates a transceiver 800, which one configuration of the transceiver 400.

[20] FIG. 9 illustrates a transceiver 900, which is another configuration of the transceiver 400.

[21] FIG. 10 illustrates a transceiver 1000, which is another configuration of the transceiver 400.

[22] FIG. 11 illustrates a transceiver 1100, which is another configuration of the transceiver 400.

- [23] FIG. 12 illustrates a transceiver 1200, which is another configuration of the transceiver 400.
- [24] FIG. 13 illustrates a communications system with a backplane interface.
- [25] Fig. 14 illustrates a transceiver 1400 with automatic polarity swap according to embodiments of the present invention.
- [26] FIG. 15 illustrates a flowchart that further describes automatic polarity swap according to embodiments of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

- [27] FIG. 1 illustrates a multi-port transceiver 100 according to embodiments of the present invention. Transceiver 100 includes a plurality of serial ports 104a-d, a plurality of parallel ports 102a-b, and a bus 106. Each of the plurality of serial ports 104 is capable of transmitting and receiving data in a serial format, and each of the parallel ports 102 is capable of transmitting and receiving data in a parallel format. For example, the serial ports 104 could be transmitting and receiving serial data with corresponding switches (e.g. MACs) 108a-d.
- [28] The transceiver 100 also includes a bus 106 that is coupled to the serial ports 104a-d and the parallel ports 102a and 102b. The bus 106 enables any serial port 104 to be connected to any other serial port 104 and to any parallel port 102 for data transmission, and vice versa. Therefore, data can be transmitted from any switch 108 to any other switch 108, or can be transmitted and received to and from the parallel ports 102. For example, data received at the serial port 104a from switch 108a can be routed to the serial port 104c by the bus 106, for transmission to the switch 108c. Additionally, data from the switch 108a can be routed to the other

serial ports 104b-d and to the parallel ports 102a and 102b through the bus 106.

[29] FIG. 2 illustrates that each serial port 104 can include a serial-to-parallel converter 202, so that serial data processed by the port 104 can be converted to parallel data, and vice versa. In other words, serial data received by serial ports 104 from corresponding switches 108 can be converted to parallel data and routed on the bus 106. The serial-to-parallel converters 202 are bi-directional, so that parallel data from the bus 106 can be converted to serial data for transmission from the serial port 104 to the corresponding switch 108. Furthermore, the parallel data from the bus 106 can also be converted to serial for transmission to the switch 108.

[30] FIG. 2 also illustrates the bus 106 to have a "ring structure" that enables the data to be sent from one adjacent port 104 to another adjacent port 104. For example, data from port 104a is directly transmitted to ports 104b and 104c over the ring structure of the bus 106. Therefore, data from any one port 104 can be connected to another port 104 using the bus 106 by transmitting data around the ring structure of the bus from one port 104 to another port 104. Furthermore, the bus 106 transmits data in a parallel format since it is connected to the parallel side of the serial to parallel converters 202. The parallel format of the bus 106 enables parallel data to be tapped out from the bus 106 at the parallel ports 102a and 102b. The ring structure of the bus 106 will be further described herein.

[31] FIG. 3 illustrates a flowchart 300 that further describes the operation of the transceiver 100 in the routing of serial data between ports. In step 302, serial data is received at a first serial port 104, from a switch 108 for example. In step 304, the serial data is converted to parallel data. In step 306, the parallel data is routed from one adjacent port 104 to another via the bus 106 until a destination port 104 is reached. In step 308, the parallel data is converted back to serial. In step 310, the serial data is

transmitted from the serial destination port 104, to a destination switch 108 for example.

[32] The ports of the transceiver 100 can be configured to operate over a number of different data standards, as will be described further herein. For example, the serial ports 104 can transmit data to the switches 108 according to an XAUI serial protocol. The XAUI serial data is converted to XGMII parallel data for transmission over the parallel bus 106, and therefore the XGMII parallel data can be tapped-out by the parallel ports 102a and 102b.

[33] FIG. 3 illustrates a flowchart 300 that further describes the operation of the transceiver 100 in the routing of data between serial ports. In step 302, serial data is received at a first serial port 104, from a switch 108 for example. In step 304, the serial data is converted to parallel data. In step 306, the parallel data is routed from one adjacent port 104 to another port 104 via the bus 106 until a destination port 104 is reached. In step 308, the parallel data is converted back to serial data. In step 310, the serial data is transmitted from the serial destination port 104, to a destination switch 108 for example. In an optional step (not shown), the parallel data can be tapped, prior to being serialized, and transmitted for further processing at another destination.

[34] Furthermore, the inverse operation can also be performed. Parallel data is received at a parallel port 102 and routed to the other parallel port 102 or routed to a serial destination port 104. If routed to a serial destination port, then the data is serialized prior to transmission.

[35] FIG. 4 illustrates a multi-port Serdes transceiver 400, which is one embodiment of the transceiver 100. The Serdes transceiver 400 includes multiple parallel ports 102 and serial ports 104, and includes the flexibility to connect any one of the parallel ports 102 to another parallel port 102 or to a serial port 104, or both. Furthermore, the multiport transceiver chip

400 can connect any one of the serial ports 104 to another serial port 104 or to one of the parallel ports 102.

[36] More specifically, the Serdes transceiver chip 400 includes two parallel transceiver ports 102a,b, and four serial transceiver ports 104a-d. However, the invention is not limited to the number of ports shown. Other configurations having a different number of ports could be used. The parallel transceiver ports 102a,b transmit and receive data in a parallel format. The parallel transceiver ports 102a,b can be XGMII parallel ports, for example, where the XGMII transceiver protocol is known to those skilled in the arts. Each XGMII port 102 can include 74 data pins, for example, operating at 1/10 the data rate of the serial ports 104. For example, the 74 pins can transmit 36 data signals and receive 36 data signals, simultaneously, and 2 clock signals (1 transmit and 1 receive).

[37] The four serial ports 104a-d can be XAUI serial ports, and transmit and receive data in a serial format. Each serial port 104 can be a quad serial port having four serial differential data lines using the XAUI protocol that is known to those skilled in the arts. In embodiments of the invention, the serial ports 104 can operate at data rates of 3.125 GHz, 2.5 GHz, and 1.25 GHz. In other words, the transceiver chip 100 is a multi-rate device. However, the XAUI data rates above are effectively quadrupled since there are four serial data lines in each serial port 104. Therefore, the 2.5GHz data rate is equivalent to a 10 GHz data rate. As discussed above, the parallel data rates can operate at 1/10 of the data rates of the serial data.

[38] The serial ports 104 can be further described as 10 Gigabit extension sub-letter (XGXS). In other words, XGXS defines the parallel-to-serial conversion between the parallel XGMII protocol to serial XAUI protocol, according to the IEEE Std 802.3ae, the entire standard of which is incorporated herein by reference. The serial ports 104 receive serial XAUI data and convert it to parallel XGMII data according to the XGXS

protocol. The parallel XGMII data is routed from adjacent port to adjacent port on the parallel bus 106.

[39] As discussed above, the parallel ports 102 and the serial ports 104 are linked together by the parallel bus 106. The parallel bus 106 enables data to travel between all the ports 102 and 104. More specifically the bus 106 enables data to travel from one parallel port 102 to another parallel port 102, and to travel from one parallel port 102 to a serial port 104. Multiplexers 402 connect the bus 106 to the parallel ports 102 and to the serial ports 104. The serial port 104 performs a parallel to serial conversion when receiving parallel data that is to be sent out serial. Likewise the bus 106 enables data to travel from one serial port 104 to another serial port 104, and to travel between a serial port 104 and a parallel port 102. The parallel port 102 enables parallel data to be tapped from the parallel bus 106 so that parallel data (e.g. XGMII data) can be transmitted from the transceiver 400. The multi-port Serdes transceiver 400 is highly flexible in being able to connect multiple parallel ports 102 to multiple serial ports 104, and vice versa.

[40] In embodiments, the Serdes transceiver chip 400 can be implemented on a single CMOS substrate. For example, the Serdes transceiver chip 400 can be implemented using a low power 0.13-micron CMOS process technology, which lends itself to higher levels of integration and application.

[41] The transceiver 400 enables dual unit operation, where one parallel port 102 is paired up with two of the serial ports 104 and the other parallel port 102 is paired up with the other two serial ports 104. For example, parallel port 102a can be paired with serial ports 104a and 104b. Likewise, the parallel port 102b can be paired with serial ports 104c and 104d. However, there is complete selectivity of the ports that are grouped together for dual unit operation. For example, parallel port 102a can be paired with either serial ports 104a and 104b, or serial ports 104c and

104d. In a backplane configuration, this provides flexibility to connect a parallel port to one or more serial ports, and with redundancy.

[42] The transceiver 400 also includes a packet bit error rate tester (BERT) 406. The packet BERT 406 generates and processes packet test data that can be transmitted over any of the serial ports 104 to perform bit error testing. Any type of packet data can be generated to perform the testing and at different data rates. For example, the packet BERT 406 can generate packet data that can be used to test the Serdes data link. As such, the packet BERT 406 provides a built-in self test for the Serdes data link. The packet BERT 406 generates test data that is sent over one or more of the serial ports 104 using the bus 106 to perform the bit error rate testing of the Serdes data link. For example, the packet BERT 406 can generate test data for a data link formed by enabling the serial ports 104a and 104b to connect the switch 108a to the switch 108c. Likewise, the packet BERT 406 can capture test data received over any one of the serial ports 104 or parallel ports 102 using the bus 106 for comparison with test data that was sent out. A bit error rate can then be determined based on this comparison.

[43] In one embodiment, the packet BERT 406 is RAM-based so that the test data is stored and compared in a RAM memory to perform the bit error rate test. In another embodiment, the packet BERT 406 is logic based so that the test data is generated by a logic function, and transmitted across a Serdes link. Upon receipt back, the test data is re-generated by the logic packet BERT 406, for comparison with the original test data that was sent over the Serdes data link. A RAM packet BERT 406 is more flexible than a logic packet BERT 406 because there is no limitation on the data that can be stored in the RAM packet BERT 406. However, a logic packet BERT 406 is more efficient in terms of substrate area because a RAM occupies more area than a logic circuit.

[44] Since the packet BERT 406 shares the same bus 106 with the serial ports 104, the packet BERT 406 can monitor (or "snoop") between the

serial ports 104. In other words, if data is being transmitted from one serial port 104 to another serial port 104, the packet BERT 406 can capture and store a portion of this data for bit error testing. In one embodiment, the packet BERT 406 "blindly" captures data being sent from one serial port 104 to another serial port 104. In another embodiment, the packet BERT 406 starts capturing data after a particular byte of data is transmitted. In another embodiment, the packet BERT 406 starts capturing data after an error event occurs. The packet BERT 406 is further described in U.S. Patent Application No. 10/681,244, filed on October 9, 2003, and incorporated by reference herein in its entirety.

[45] The Serdes transceiver chip 400 also includes the ability to include other optional logic blocks 408 that are not necessary for the operation of the Serdes transceiver. In other words, the logic blocks 408 could be customer driven logic blocks or some other type of logic block. These optional logic blocks 408 can transmit and receive data over the serial ports 104 or parallel ports 102 using the bus 106. The packet BERT 406 and the optional blocks 408 connect to the bus 106 using the multiplexers 404.

[46] The Serdes transceiver chip 400 also includes a management interface 412 that enables the configuration of the portions (parallel ports 102, series port 104, packet BERT 406, and optional logic blocks 408) of the transceiver chip 100. The management interface 412 includes two pads 414 that enable two different management chips to program and control the portions of the transceiver chip 400 using MDIOs blocks 416. For example, one management chip connected to pad 414a could control the parallel port 102a and the serial ports 104a and 104b, and another management chip connected to pad 414b could control the parallel port 102b and the serial ports 104c and 104d. The management interface 412 is configured to be compatible with both IEEE Std. 802.3 clause 45 and the IEEE Std. 802.3 clause 22 management standards. In other words, one management pad 414a and MDIO block 416a can be programmed to be

responsive to clause 45 electricals and protocol, and the other management pad 414b and MDIO block 416b could be responsive to clause 22 electricals and protocol. Furthermore, the management pads 414 and MDIO blocks can mix and match clause 22 and clause 45 electrical and protocols. For example, management pad 414a and MDIO block 416a can be responsive to clause 22 electricals and clause 45 protocols, and vice versa. Similar mix and match can be done for the management pad 414b and the MDIO block 416b. The management data pads are further described in U.S. Patent Application No. _____, titled "Multipurpose and Integrated Pad Ring for Integrated Circuit", filed herewith, Attorney Docket No. 1875. 4520000, and U.S. Patent Application No. _____, titled "Programmable Management I/O Pads for an Integrated Circuit", filed herewith, Attorney Docket No. 1875. 4530000, both of which are incorporated by reference herein in its entirety.

[47] FIG. 5 illustrates the substrate layout 500 for the Serdes transceiver 400 according to embodiments of the invention. The substrate layout 500 is configured to minimize the substrate area of the transceiver 400, and efficiently provide the port interconnections described above.

[48] The substrate layout 500 is configured so that the parallel ports 102a,b and the serial ports 104a-d are on the outer perimeter of the substrate 200, as shown. In other words, the serial ports 104a and 104b are on a first side of the substrate layout 500 and the serial ports 104c and 104d are on a second side of the substrate layout 500. The parallel port 102a is on a third side of the substrate layout 500. And the parallel port 102b is on a fourth side of the substrate layout 500. A logic core 502 is at the center of the substrate 500, where the logic core 502 operates the bus 106 and the serial 104 and parallel 102 data ports. The management interface 412, the packet BERT 406, and the optional logic blocks 408a-c are adjacent to the logic core 502 as shown. The bus 106 can be described as a "ring" structure (or donut "structure") around the logic core 502, and is placed in between the logic core 502 and the data ports 102 and 104 that

occupy the parameter of the substrate layout 500. Furthermore, the ring structure of the bus 106 also provides efficient communication between the logic core 502 and the various data ports 102 and 104. Furthermore, the ring structure of the bus 106 also provides efficient communication between the management interface 412, the packet BERT 406, the optional logic blocks 408, and the various data ports 102 and 104.

[49] The bus 106 is illustrated as 8 sections 106a-106g for ease of illustration. Each section provides an interface to the respective data ports 102 or 104 that are adjacent to the respective sections.

[50] FIG. 6 represents one of the 8 sections 106a-106g of the bus 106 according to embodiments of the present invention. Each section of the bus 106 can be represented as two paths 608 and 610. Data enters the bus 106 through a buffer 602 and proceeds to its destination along the path 608 and through the buffers 604. Data passes from one section to another section of the bus 106 using the path 610 and passing through the buffers 612. The mux 606 represents data passing from the bus 106 to a functional block, such as a data port 102, 104 or the packet BERT 406. The actual wires and buffers on the bus 106 are matched to minimize signal distortion.

[51] In embodiments, the data wires in the bus 106 are deposited on the substrate 500 in a particular fashion. Namely, a power or ground is placed between adjacent (or near by) data wires. Furthermore, adjacent data wires on the bus 106 are placed on two separate layers. Therefore, a power or ground will be above or below a data wire, and adjacent to a data wire. Therefore, two nearby data wires will not be located directly adjacent one another, but instead will be positioned diagonal to each other, thereby reducing cross talk.

- [52] The parallel bus 106 is further described in U.S. Patent Application No. _____, titled "Cross-Linked Multiplexer Bus", filed herewith, Attorney Docket No. 1875. 3640002, which is incorporated by reference herein in its entirety.
- [53] The multi-port Serdes transceiver 400 supports multiple different data protocols at the pads including XGMII, TBI, RTBI, HSTL, SSTL, or LVTTTL protocols, and others.
- [54] FIG. 7 further illustrates an example layout of the bus 106. The wires 702 between data ports 102, 104 are configured to have the same path lengths so as to minimize signal distortion. In other words, wires 702a-d are deposited so as to have the same path length so as to reduce signal distortion.
- [55] The multi-port Serdes transceiver 400 includes the ability to change the timing of the data ports 102 and 104. This includes the ability to change the timing between the data and clock signals. In other words, the registers in the data ports 102 and 104 can be re-programmed to operate at different timing protocols.
- [56] The Serdes transceiver 400 is highly flexible and can be configured to provide multiple different transceivers by enabling and disabling various serial and parallel data ports. This is accomplished using a configuration logic circuit 418 that controls the registers and timing of the data ports 102 and 104, and also enables/disables these data ports. As a result, several different transceiver products can be configured from a single semiconductor die that is manufactured with the 4 serial data ports and 2 parallel data ports. For instance, FIG. 1 illustrates an embodiment, where all four serial ports 104 and both parallel ports 102 are enabled and accessible to exchange data. The transceiver 400 represents the maximum capability that can be achieved from the 4 serial and two parallel data ports. Other specific examples are discussed below, but the invention is not limited to these examples.

[57] FIG. 8 illustrates a transceiver 800 having 4 serial XGXS data ports 104. The two parallel ports 102 have been disabled using configuration logic 418 and/or by not bonding the parallel ports to the output pins of the semiconductor package. The transceiver 800 enables any of the switches 108 to exchange data with any of the other switches 108.

[58] FIG. 9 illustrates a transceiver 900 having 2 serial XGXS data ports 104a and 104b and two XGMII parallel data ports 102a and 102b. The other two serial data ports 104c and 104d have been disabled using configuration logic 418 and/or by not bonding the ports to the output pins of the semiconductor package. The transceiver 900 enables the switch 108a to exchange data with the switch 108b, and allows the parallel XGMII data to be tapped from the parallel bus 106 using the parallel XGMII data ports 102a 102b.

[59] In another embodiment of FIG. 9, the timing of the serial and parallel ports is varied to provide another transceiver product. For instance, the two XGXS serial ports can be configured as 8 Serdes ports that operate at a slower data rate than the XGXS data ports. Likewise, the two XGMII parallel data ports can be configured as 8 TBI data ports that also operate at a slower data rate. In general, the timing of the data ports can be revised and reconfigured for each of the transceiver configurations shown herein.

[60] FIG. 10 illustrates a transceiver 1000 having 2 serial XGXS data ports 104a and 104b and only one XGMII parallel data port 102a. The other two serial data ports 104c and 104d and the parallel port 102b have been disabled using configuration logic 418 and/or by not bonding the ports to the output pins of the semiconductor package. The transceiver 1000 enables the switch 108a to exchange data with the switch 108b, and allows the parallel XGMII data to be tapped from the parallel bus 106 using the parallel XGMII data port 102a.

[61] FIG. 11 illustrates a transceiver 1100 having 2 serial XGXS data ports 104a and 104c and no XGMII parallel data port 102a. The other two

serial data ports 104c and 104d and the parallel ports 102 have been disabled using configuration logic 418 and/or by not bonding the disabled ports to the output pins of the semiconductor package. The transceiver 1100 enables the switch 108a to exchange data with the switch 108c.

[62] FIG. 12 illustrates a transceiver 1200 having 3 serial XGXS data ports 104a, 104b, and 104d and one XGMII parallel data port 102a. The other serial data ports 104c and the parallel port 102a have been disabled using configuration logic 418 and/or by not bonding the disabled ports to the output pins of the semiconductor package. The transceiver 1200 enables the switches 108a, 108b, and 108d to exchange data with each other.

[63] Based on the above discussion, it is apparent that the base transceiver 400 is highly flexible and enables multiple transceiver products, with different capabilities and price points, to be configured and sold from the base semiconductor die.

Automatic Polarity Swap

[64] FIG. 13 illustrates a communications system 1300 having a first Serdes transceiver 100a that communicates data with a second Serdes transceiver 100b through an interface 1303, that may be a backplane for example. The Serdes transceivers 100a and 100b can be any type of the transceivers discussed herein, or any other type of transceiver. The Serdes transceiver 100a can be represented as having a differential input 1301p and 1301n and a differential output 1302p and 1302n. Likewise, the Serdes transceiver 100b can be described as having a differential input 1304p and 1304n, and a differential output 1306p and 1306n.

[65] During the configuration of the backplane 1303, the differential output 1302 of the transceiver 100a can be mistakenly cross-connected with the differential input 1304 of the Serdes transceiver 100b. In other

words, the 1302n output of the transceiver 100a can be mistakenly connected to the 1304p input of the transceiver 100b. Likewise, the 1302p output of the transceiver 100a can be mistaken connected to the 1304n input of the transceiver 100a. The result of such a cross-connection is that invalid data words are received at the Serdes transceiver 100b, which causes an increase in the bit error rate (BER).

[66] FIG. 14 illustrates an apparatus and method of automatic polarity swap to address the cross-connection problem discussed with reference to FIG. 13. FIG. 14 illustrates a transceiver 1400 having an error check and correction module 1401. The error check and correction module 1401 includes an error check 1402 and a exclusive OR (X-OR) 1404. The error check and correction module 1401 is coupled to the output of the serial-to-parallel converter 202 and checks the digital output of the serial-to-parallel converter 202 to determine if it is a valid digital word. If it is not a valid digital word, then the module 1401 performs an automatic polarity flip of the digital word.

[67] The error check and correction module 1401 includes an error check 1402 at the output of the parallel-to-serial converter 202a. The error check module 1402 examines the digital word at the output of the parallel-to-serial converter 1402 and determines if it is a valid digital word within the context of the communications system 1300. For example, the error check module 1402 can include a RAM memory that stores the possible digital word combinations for comparison with the digital word output of the serial-to-parallel converter 202. If the digital word output of the serial-to-parallel converter 202a does not match any one of the possible combinations, then the digital word output of the serial-to-parallel converter is not a valid digital word. Accordingly, one possibility is that the interface 1303 is cross-connected as discussed above. If this is so, then flipping the polarity of the bits that make up the digital word, will convert the invalid digital word to a valid digital word. This can be accomplished using an exclusive-OR gate 1404. When the error check module 1402

detects an invalid digital word, then the error check module 1402 sends a control bit logic "1" to the exclusive-OR 1404, which causes the exclusive OR to invert the digital output from the serial-to-parallel converter, and generate a inverted digital word 1405. In other words, the logic state of each portion (both p and n) of the differential line is inverted from 0 to 1, or 1 to 0. If the error was introduced by the cross-connection, then the inverted digital word 1405 will be a valid digital word.

[68] The error check module 1402 outputs a control bit "0" when it determines that the digital word at the output of the serial-to-parallel converter is a valid digital word. A control bit "0" does not invert the digital word, so that the digital word passes unchanged for further processing.

[69] The error check and correction module 1401 can be implemented using other configurations as will be understood by those skilled in the arts based on the discussion given herein. For instance, a logic circuit other than an X-OR can be used to flip the polarity of the digital word if it is invalid.

[70] An advantage of implementing the automatic polarity swap is that cross-connect errors are rectified quickly and easily, without having to re-wire or re-configure any hardware. Furthermore, the polarity swap can be implemented on the transmit side or the receive side. However, the receive implementation is shown in FIG. 14. The transmit implementation will be apparent based the description related to FIG. 14.

[71] FIG. 15 illustrates a flowchart 1500 that further describes an automatic polarity swap according to embodiments of the present invention.

[72] In step 1502, a serial differential signal is received. In step 1504, the serial differential signal is converted to a parallel differential data word.

- [73] In step 1506, the parallel differential data word is examined to determine if it is a valid data word for the corresponding communications system. For instance, the parallel differential data word can be compared with valid data words that are stored in a RAM. If the parallel differential data word is valid, then it is passed unmodified for further processing. For example, a valid data word can be re-serialized and transmitted to a destination switch or MAC.
- [74] In step 1508, the parallel differential data word is inverted if the parallel differential data word is found to be invalid in step 1506. The differential data word includes a first component and a second component (i.e. positive component and a negative component), and the first component and the second component are both inverted so that a "0" becomes a "1", and a "1" becomes a "0". For instance, a logic circuit (such as the X-OR 1404) can be used to invert the parallel data word if it is invalid.
- [75] In step 1510, the inverted parallel data word can be re-examined to determine if it is now a valid data word, and if so the inverted parallel data word can be further processed. For example, the inverted parallel data word can be serialized and retransmitted to a destination switch, or MAC.

Conclusion

[76] Example embodiments of the methods, systems, and components of the present invention have been described herein. As noted elsewhere, these example embodiments have been described for illustrative purposes only, and are not limiting. Other embodiments are possible and are covered by the invention. Such other embodiments will be apparent to persons skilled in the relevant art(s) based on the teachings contained herein. Thus, the breadth and scope of the present invention should not be limited by any of the above-described exemplary embodiments, but should be defined only in accordance with the following claims and their equivalents.